What is claimed is:

1. A mapping algorithm for transforming models between the two types, from a Forrester flow diagram (FD) to a partial recurrent neural network (PRN), and vice versa, the mapping algorithm comprising:
   relating **levels** (and **constants**) to the input, output, and state units;
   relating **rates** (and **auxiliaries**) to hidden units;
   relating **wires** to links from the said state units to the said hidden units;
   relating **flows** to links from the said hidden units to the said output units;
   assigning the value of **DT** as the weights of links from the said hidden units to the said output unit; and
   assigning coefficients in rate equations as the weights of links from the said state units to the said hidden units.
2. A semiautomatic learning method for system dynamics model (SDM) construction and manipulation, the method comprising:
   creating an initial structure of the said PRN;
   creating a training set with special arrangements; and
   training the said PRN with the said training set.
3. A policy design method for SDMs, the method comprising:
   representing a target SDM as said PRN;
   according to the intention of a model constructor, training said PRN with a special arrangement data set like those in said claim 2; and
   identifying the changes in structure and parameters values between the two said PRNs, which leads to an overall policy for model manipulation.
4. The mapping algorithm, as recited in claim 1, further comprising:
   implementing a level equation by a weighted sum of output values from said hidden and said state units connected to said output unit via links;
   implementing a rate equation by a weighted sum of output values from said state units connected to said hidden unit via links;
   relating initialization equations to the corresponding links from said input units to said output units; and
   relating constant equations to said corresponding links from said state units to said output units, and also from said output units to said state units.
5. The method of claim 2, wherein said step of creating a training set with special arrangement including:
   creating a set of two-part training tuples, with the input part representing values for said input units and the output part representing values for said output units;
   assigning both of the two parts of said first training tuple with the initial values

of levels and constants;

assigning the output part of the rest of said training tuples with the historical time series of data from said levels and constants, with one tuple for each time step;

resetting the input part of the rest of said training tuples to zero; and

ordering said training tuples in time sequence.

6. The mapping algorithm, as recited in claim 1, further comprising:

   interpreting the structure of said PRN learned by said method of claim 2 and transforming it back to said FD using the relationships listed in claim 1; and

   dropping those links from said state units to said output units with near-zero weights.

7. The method of claim 3, wherein said step of training said PRN with a special arrangement data set including:

   using a flat line as the training data set if the problem is to search for a policy that will generate a stable trajectory for a given model; and

   generating the training data set either by an optimal algorithm or manually by a domain expert if the problem is to search for a policy that will generate a growing trajectory for a given model.